ub=n-1;

ARRAYS

1 What is an array?

① · An array is a collection of ~~variables~~ values of the same data type that are referenced by a common name.

② · It is a collection of similar type of elements that have contiguous memory location.

· It is a data structure where we store similar elements.

· We can store only fixed number of elements in an array.

Declaration of an array

We must declare a variable to reference the array and we must specify the type of variable the array can reference.

dataType arrayVariable []; ⇒ int a[];
declares array will contain values of 'int' type
double b[];

Creating an array

We can create an array by using the new operator.

int a[];
a = new int [10]; → allocates 10 memory locations to array 'a'.

This does 2 things:-

· It creates an array using new dataType [arraySize];

· It assigns the reference of the newly created array to the variable arrayVariable.

Initializing an array: (means assigning value to array elements).

An array element is addressed by using the array name followed by a integer subscript in brackets. arrayname [index] = value; OR

We can declare, create and initialize as

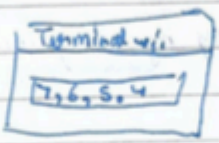
type arrayName [] = {value0, value1, value2 ..., value n-1};

How to differentiate array variable from ordinary variable
Ordinary variable contains only one value in the memory.
Array variables contains a number of variables in the memory.

How to store data in array

i) By using assignment statement (direct initialization of array)
int a[] = {1,2,3,4};
OR
char p[] = {'a','b','c'};
↓ entry of elements straightaway at time of its declaration.

ii) By using BlueJ system
void main (int m[])
{
for(i=0; i<5; i++) in window
{
6



iii) By using Input Stream

```
for(i=0; i<n; i++)
{
    a[i] = Integer.parseInt(in.readLine());
}
```

Cell number of array element

Subscript or Index

1. The cell number of the array element used under square bracket [] is said to be the subscript.
2. The element no. in [] that refers to the data element being accessed is called subscript or index.
3. The subscript of an element designates its position in the array sequence.
   a[5] refers to element of array 'a' at 5th subscript or 6th element.
4. Subscripts start from zero and go on till size − 1.

Subscripted variable = An array variable is also called subscripted variable because it is compound of many subscripts and provides access to more than one memory location and its elements.

[m[4];
4 = subscript m = subscripted variable]

Position of an element is [index + 1].

Out of bounds subscripts =
The subscripts other than 0 to n−1 (both limits inclusive) for an array having 'n' elements are called out-of-bounds subscripts.

out-of-bounds Exception
Reference to out-of-bounds subscript produces this exception and the program will crash if exception is not handled.

- Subscript of 1st data item = zero
- num[9] refers to 10 element of array 'num'
- p[n] refers to (n+1)th element.
- last ele no. of array = (n-1)
- char p[5] = sixth character of array p
- int p[5] = sixth integer of array p

Length VS length() instance variable VS method

The length is an instance variable of array and it returns the number of elements in that array.

length() is a method of String class and returns the length of the string.

length is a property of array that is defined in JVM.
length() is a method defined in String class. to count no. of ch

String abc = "Sample";
abc.length() → 6

int aby = {2,3,4};
aby.length → 3

Memory used = (size of data type x size of array) bytes
Eg. char A[4] = 2×4 = 8 bytes

Show example

MEMORY REPRESENTATION OF SDA

Single dimensional arrays are essentially lists of information of the same type and their elements are stored in continuous memory locations in their index order.

Arrays are stored as special objects containing :-

- a group of contiguous memory locations that all have the same name and same data type.
- A reference that stores the beginning address of the array elements.
- A separate instance variable containing the number of elements in the array.

Initializing arrays

- When an array object is created with the new operator, its elements are automatically initialized to ~~boolean~~ zero, which is the default initial value of all numeric types.
- If array is boolean, all elements are initialized with boolean false value
- We can initialize arrays to non zero values using array initializers - which is comma separated list enclosed in braces.

```
int a[] = {20, 30, 40};
```

Arrays are treated as objects as they are reference types.

```
int a1[] = {2,3,5,7}; int a2[] = {2,3,5,7};
a1 == a2 // false as '==' compares references, not values
a1.equals(a2) // false as it does not compare data.
a1 = a2 makes them refer to same object
```

TYPES OF ARRAY

1) Single dimensional array comprised of finite homogenous elements. When elements are specified by a variable with a single subscript, the array is called single dimensional array.
   age [5];

2) Double dimensional array comprised of elements, each of which is itself an array. It is a structure created in memory to contain data values by using rows and columns.
   It has 2 subscripts - one for row number and other for column number.

ARRAY DEFINITION

An array must be defined before it can be used to store information.
Array definition specifies :-

1) a base type (variable type) = data type of the array elements
2) Array name = The name with which the array will be referenced.
3) size = how many data items the array will contain.
   must be an integer value or integer constant without any s

BASIC OPERATIONS ON AN ARRAY

a) Searching
b) Sorting
c) Merging
d) insertion
e) deletion

Need of array concept

- A single variable can store only one value in its single memory location.
- If attempted to store another value, the previous value gets overwritten.
- one can use more than one variables but this reduces the quality of program
- Therefore, array concept is needed.

USE OF ARRAYS

Arrays are useful where many elements of the same type need to be stored and processed. Uses are:-

- To group storage locations.
- To declare elements in a brief manner using subscripts.
- To store elements in continuous locations.

ADVANTAGES

1) Easy to specify = The declaration, allocation of memory and initialization can all be done in one line.

2) Free from run-time overheads = There is no run-time overhead to allocate memory apart from once at the start and once at the end.

3) Random access of elements:
Array facilitates random direct access to any element via its subscript.

4) Fast sequential access
It is faster to sequentially access elements due to contiguous storage and constant computation of the address of a component.

5) Simple code
Multiple data items of same type can be referenced under a common name.

DISADVANTAGES

1) Need to know the size at time of allocation.

2) Careful design regd. to make sure array will be large enough to hold the largest possible group of data but no larger.

3) Not suitable for situations demanding varying memory sizes.
It may either lead to shortage of memory or wastage of memory.

Difference between insertion and deletion
Insertion is process of adding a new element in an array at a particular place.

The process of deletion is the removal of an element from a particular place in the array.

SEARCHING

Searching is the process to determine whether the given item is present in the array or not and returning its index (if present).

1) Linear search (Sequential search)

- Linear search refers to the search technique in which each element of the array is compared with the search item one by one, until the search item is found or all elements have been compared.
- Firstly, element at 0th index is compared, then at 1st index and so on till (n-1)th index.
- This method traverses the array sequentially to locate the given item.
- After completion, a relevant message is displayed to confirm that search is completed successfully or unsuccessfully.

Advantage = Works on unsorted arrays as well.

Disadvantage = requires more comparisons than binary search in large arrays

Linear search	Binary search
• can work on both sorted and unsorted arrays	Works only on sorted arrays
• Search begins at start of array (0th element) and continues till a match is not found.	Array is divided into 2 halves and the the search element is searched either in 1st half or 2nd half.
• becomes inefficient if array size increases.	is more efficient in case of larger arrays.

2) Binary search

- Binary search is a search technique that works only for sorted arrays.
- Here the search item is compared with the middle element of the array.
- If the search item matches with the element, the search finishes.
- If search item is less than middle (in ascending order), binary search is performed in the 1st half of the array (lower half), otherwise binary search is performed in latter half of array.
- Search segment reduces to half at every successive stage.
- The process is repeated until either the item is found or the array segment is reduced to single element.

Pre-conditions of binary search
1) Array must be sorted. ① sorted,
2) Lower bound, upper bound and sort order of array must be known. ② limit + order

Efficiency of binary VS Linear
- Linear search compares search item with each element of array one by one.
If search item is in beginning of array, comparisons are low.
If element is in the last, this technique proves worst as so many comparisons take place.
- Binary search tries to locate element in minimum possible comparisons provided the array is sorted.
- This technique proves efficient in nearly all cases.

arranging array elements in a specified order

SORTING,

Sorting means arranging the array elements in a specified order i.e, either ascending, descending or alphabetical order depending upon the values of the elements.

1) Selection sort

① It is a sorting technique where the next smallest (or next largest) element is found in the array and is moved to its correct position.

② It is based on pick and exchange technique.

③ If array is to be arranged in ascending order, the smallest element needs to be selected and exchanged with the 1st element of the array.

④ Again the smallest element from remaining elements is selected and exchanged with second array element.

⑤ This process continues till all elements are arranged.

No. of comparisons $= n(n-1)/2 = n(n-1)/2$

```
for(i=0 ; i< size-1 ; i++)
{
    minpos = i;
    for(j=i+1; j< size ; j++)
    {
        if(a[j] < a[minpos])
        {
            minpos = j;
        }
    }
    t = a[i];
    a[i] = a[minpos];
    a[minpos] = a[i];
}
```

→ minpos

Swap i, minpos

consecutive elements

Bubble sort

1. In this technique, consecutive elements are compared seperately and exchanged if they are not in correct order.
2. This process will continue till all elements are arranged in desired order.
3. The idea is to move the largest element to the highest index position in the array.

No. of iterations $= \frac{n(n-1)}{2}$

8 elements $\Rightarrow \frac{7 \times 8}{2} = 28$

```
for (i=0; i<n-1; i++)
{
    for (j=0; j<n-1-i; j++)
    {
        if (a[j] > a[j+1])
        {
            temp = a[j];
            a[j] = a[j+1];
            a[j+1] = temp;
        }
    }
}
```